US 20030195917A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0195917 A1**

Horiyama et al. (43) **Pub. Date: Oct. 16, 2003**

(54) **MANAGEMENT SYSTEM, INFORMATION PROCESSING APPARATUS, AND MANAGEMENT METHOD AND PROGRAM**

(75) Inventors: **Jun Horiyama**, Kanagawa (JP); **Koji Nozato**, Kanagawa (JP)

Correspondence Address:
FITZPATRICK CELLA HARPER & SCINTO
30 ROCKEFELLER PLAZA
NEW YORK, NY 10112 (US)

(73) Assignee: **Canon Kabushiki Kaisha**, Tokyo (JP)

(21) Appl. No.: **10/405,222**

(22) Filed: **Apr. 3, 2003**

(30) **Foreign Application Priority Data**

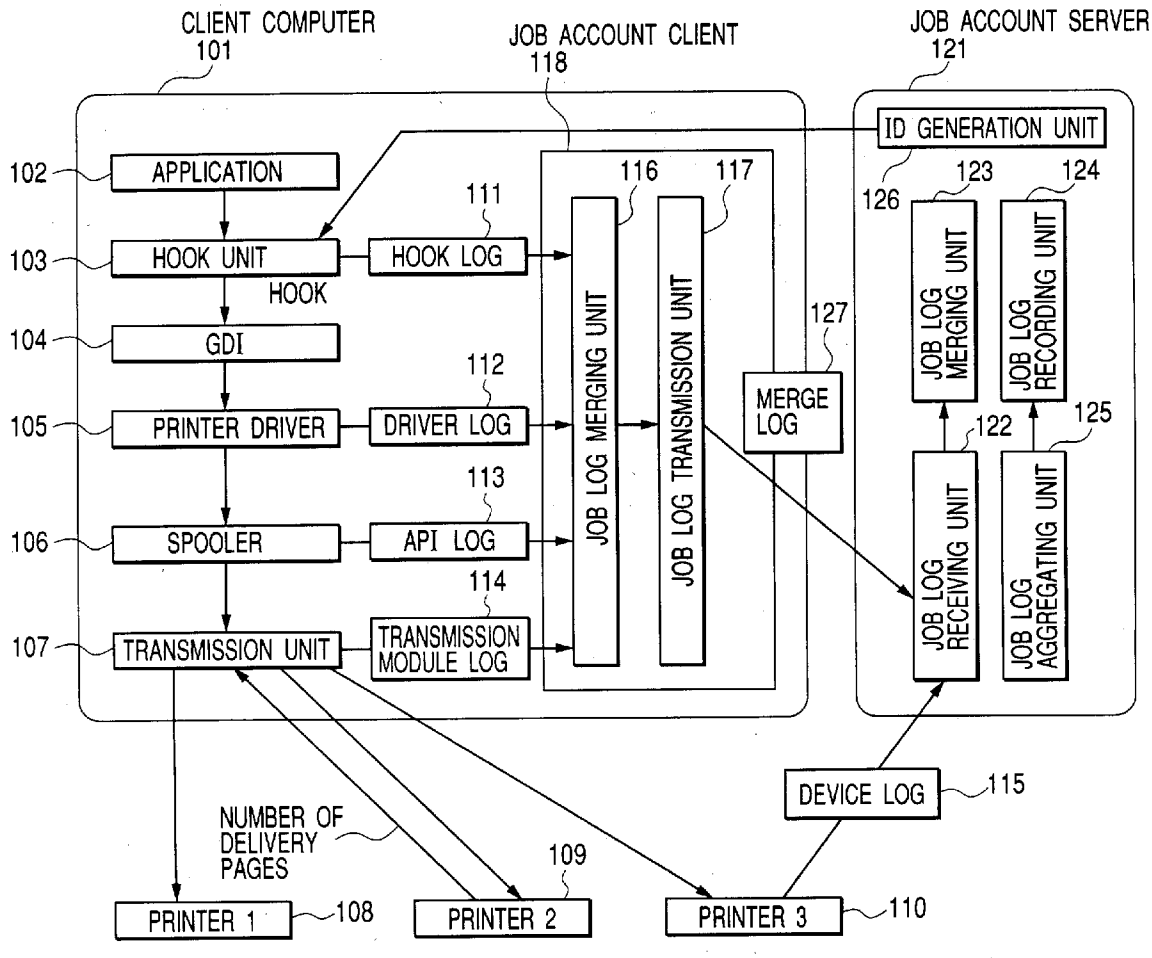
Apr. 12, 2002 (JP) 110687/2002
Apr. 12, 2002 (JP) 110688/2002

Publication Classification

(51) **Int. Cl.**[7] **G06F 9/00**
(52) **U.S. Cl.** **709/102**

(57) **ABSTRACT**

The present invention makes it possible to uniquely manage the use of peripherals over a wide range or in two or more bases. The present invention controls whether all job information, managed by a management apparatus, is acquired from the management apparatus managing information on data processing, performed by a device, as job information, or additional job information among job information managed by the management apparatus is acquired.



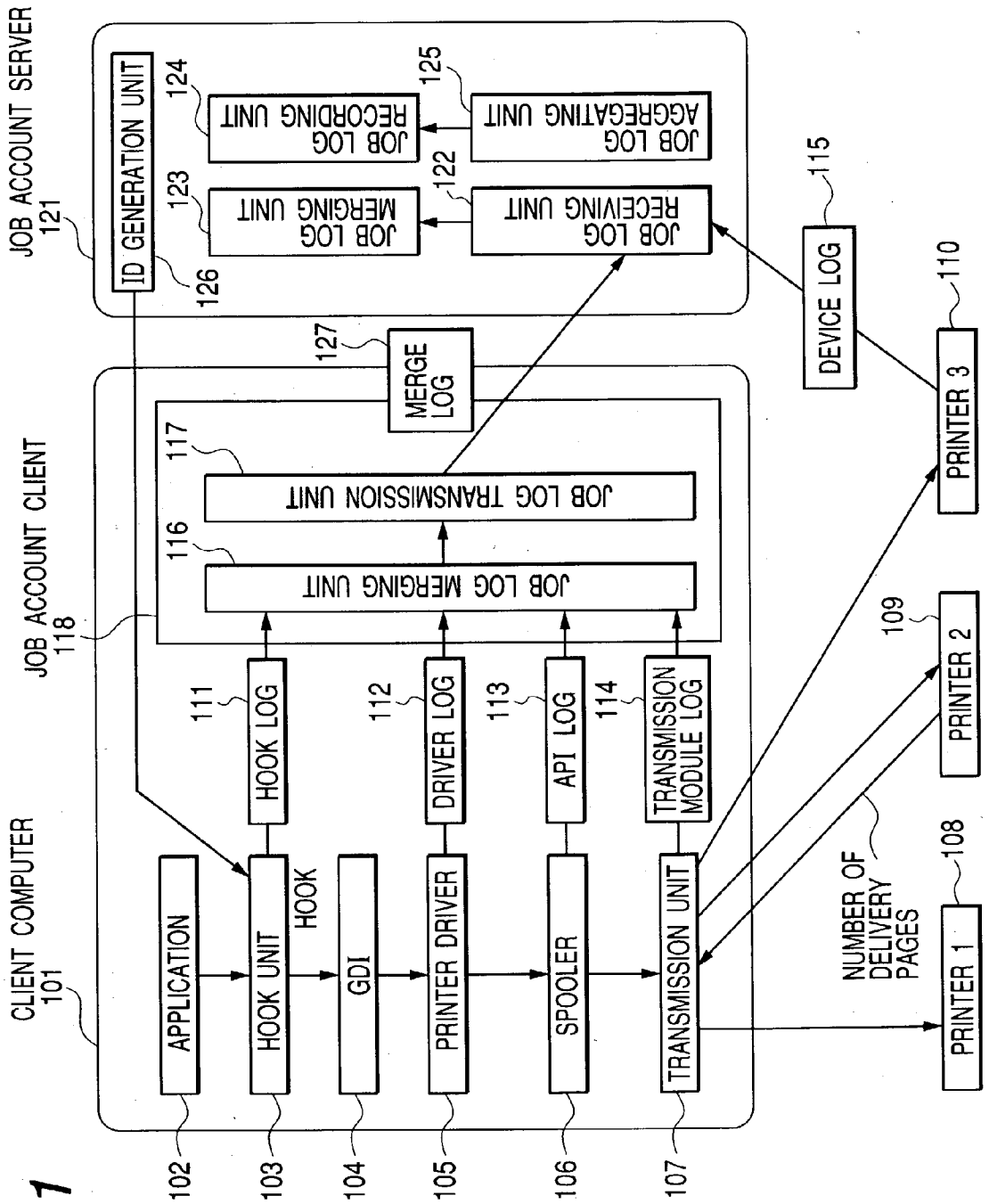

CLIENT COMPUTER
101
APPLICATION
HOOK UNIT
HOOK
GDI
PRINTER DRIVER
SPOOLER
TRANSMISSION UNIT
JOB ACCOUNT CLIENT
118
HOOK LOG
DRIVER LOG
API LOG
TRANSMISSION MODULE LOG
JOB LOG MERGING UNIT
JOB LOG TRANSMISSION UNIT
MERGE LOG
JOB ACCOUNT SERVER
121
ID GENERATION UNIT
JOB LOG RECORDING UNIT
JOB LOG AGGREGATING UNIT
JOB LOG MERGING UNIT
JOB LOG RECEIVING UNIT
DEVICE LOG
115
PRINTER 1
PRINTER 2
PRINTER 3
108
109
110
NUMBER OF DELIVERY PAGES
102
103
104
105
106
107
111
112
113
114
116
117
122
123
124
125
126
127

FIG. 1

FIG. 2

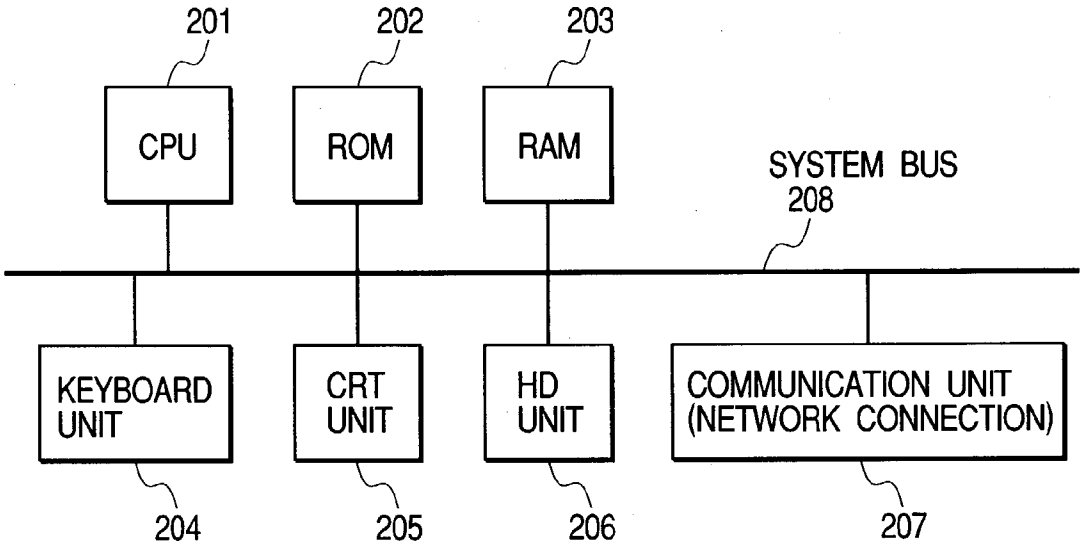

201
202
203
CPU
ROM
RAM
SYSTEM BUS
208
KEYBOARD UNIT
CRT UNIT
HD UNIT
COMMUNICATION UNIT (NETWORK CONNECTION)
204
205
206
207

FIG. 3

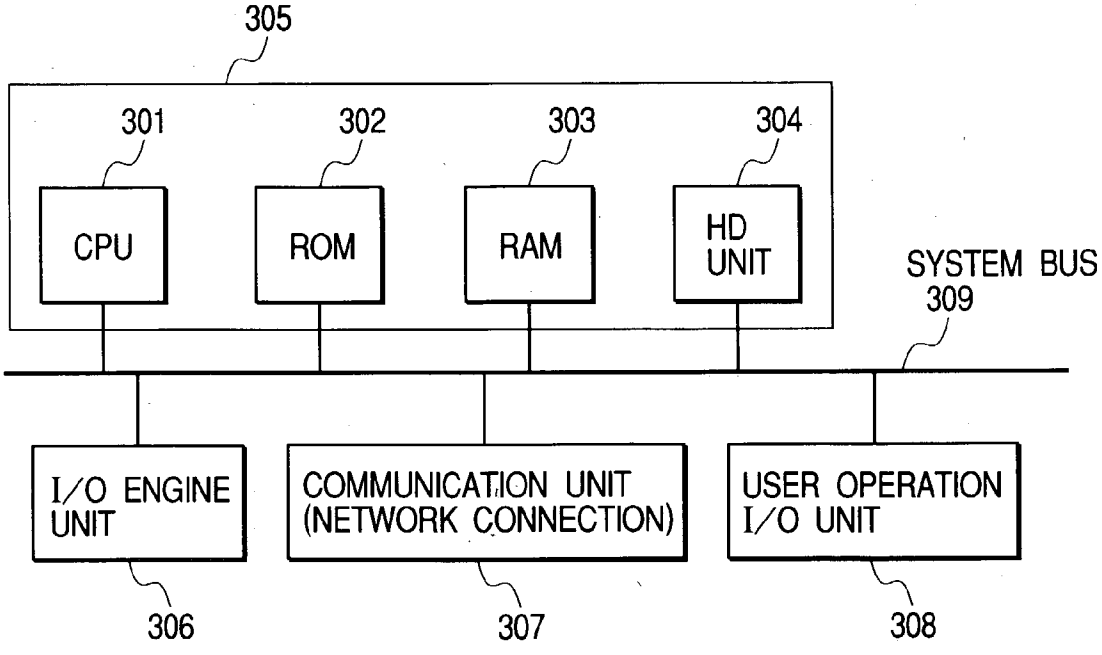

305
301
302
303
304
CPU
ROM
RAM
HD UNIT
SYSTEM BUS
309
I/O ENGINE UNIT
COMMUNICATION UNIT (NETWORK CONNECTION)
USER OPERATION I/O UNIT
306
307
308

FIG. 4

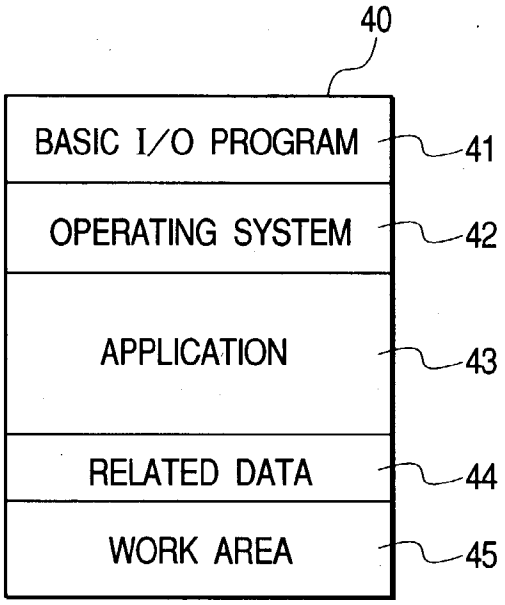


FIG. 5

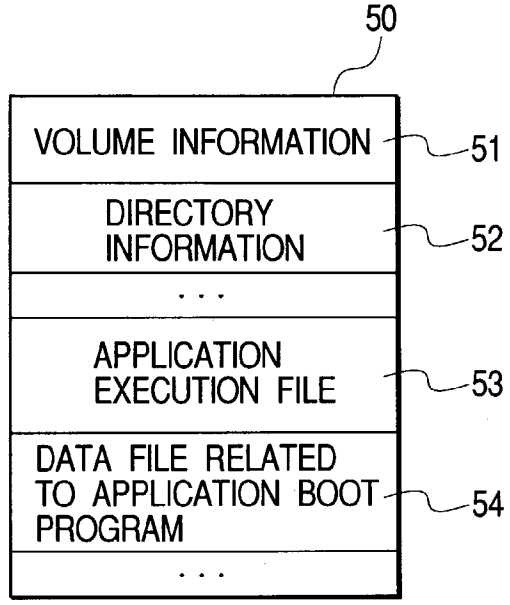


FIG. 6

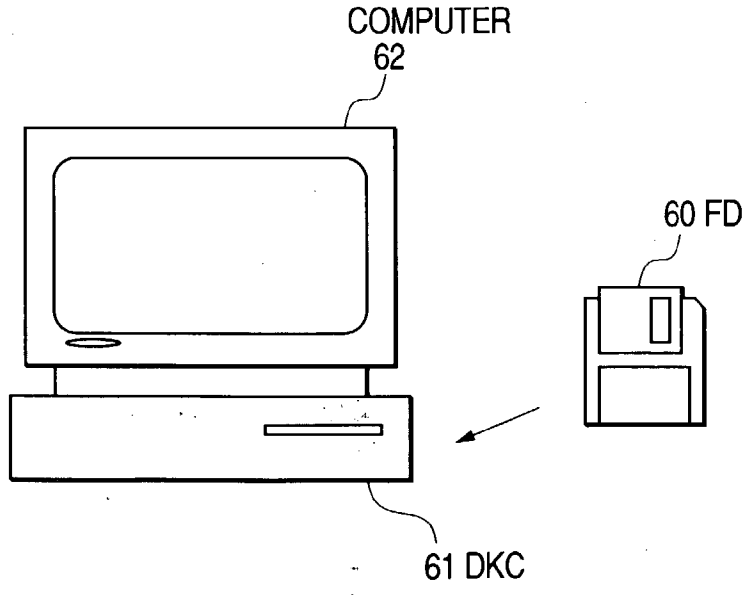


FIG. 7

JobId	520
APPLICATION NAME	WORD MEIJIN (EXPERT)
NUMBER OF LOGICAL PAGES	9
DOCUMENT NAME	abcd0001
ORIGINAL DOCUMENT NAME	ANNUAL REPORT IN FISCAL YEAR 2001
PROCESSED	NO
TYPE	HOOK
START TIME	2001/12/20 21:11
END TIME	2001/12/20 21:21

FIG. 8

JobId	520
PAPER SIZE	A4
N-up	4
NUMBER OF PHYSICAL PAGES	3
DUPLEX/SIMPLEX	DUPLEX
PROCESSED	NO
TYPE	DRIVER
START TIME	2001/12/20 21:21
END TIME	2001/12/20 21:31

FIG. 9

JobId	520
OWNER'S NAME	YAMADA
SPOOL DATA SIZE	56789
NUMBER OF PHYSICAL PAGES	3
DUPLEX/ SIMPLEX	DUPLEX
PROCESSED	NO
TYPE	API
START TIME	2001/12/20 21:31
END TIME	2001/12/20 21:41

FIG. 10

JobId	520
PROCESSED	NO
TYPE	TRANSMISSION MODULE
START TIME	2001/12/20 21:41
END TIME	2001/12/20 21:51

FIG. 11

JobId	520
APPLICATION NAME	WORD MEIJIN (EXPERT)
NUMBER OF LOGICAL PAGES	9
DOCUMENT NAME	abcd0001
ORIGINAL DOCUMENT NAME	ANNUAL REPORT IN FISCAL YEAR 2001
PROCESSED	NO
TYPE	MERGING
START TIME	2001/12/20 21:11
END TIME	2001/12/20 21:21
PAPER SIZE	A4
N-up	4
NUMBER OF PHYSICAL PAGES	3
DUPLEX/SIMPLEX	DUPLEX
OWNER'S NAME	YAMADA
SPOOL DATA SIZE	56789

FIG. 12

JobId	520
NUMBER OF DELIVERY PAGES	3
NUMBER OF DELIVERY SHEETS	2
PROCESSED	NO
TYPE	TRANSMISSION MODULE
START TIME	2001/12/20 22:01
END TIME	2001/12/20 22:11
DOCUMENT NAME	abcd0001
DEVICE NAME	XYZ

FIG. 13

JobId	520
APPLICATION NAME	WORD MEIJIN (EXPERT)
NUMBER OF LOGICAL PAGES	9
DOCUMENT NAME	ANNUAL REPORT IN FISCAL YEAR 2001
PROCESSED	NO
TYPE	MERGING
START TIME	2001/12/20 21:11
END TIME	2001/12/20 21:21
PAPER SIZE	A4
N-up	4
NUMBER OF PHYSICAL PAGES	3
DUPLEX/SIMPLEX	DUPLEX
OWNER'S NAME	YAMADA
SPOOL DATA SIZE	56789
NUMBER OF DELIVERY PAGES	3
NUMBER OF DELIVERY SHEETS	2
DEVICE NAME	XYZ

FIG. 14

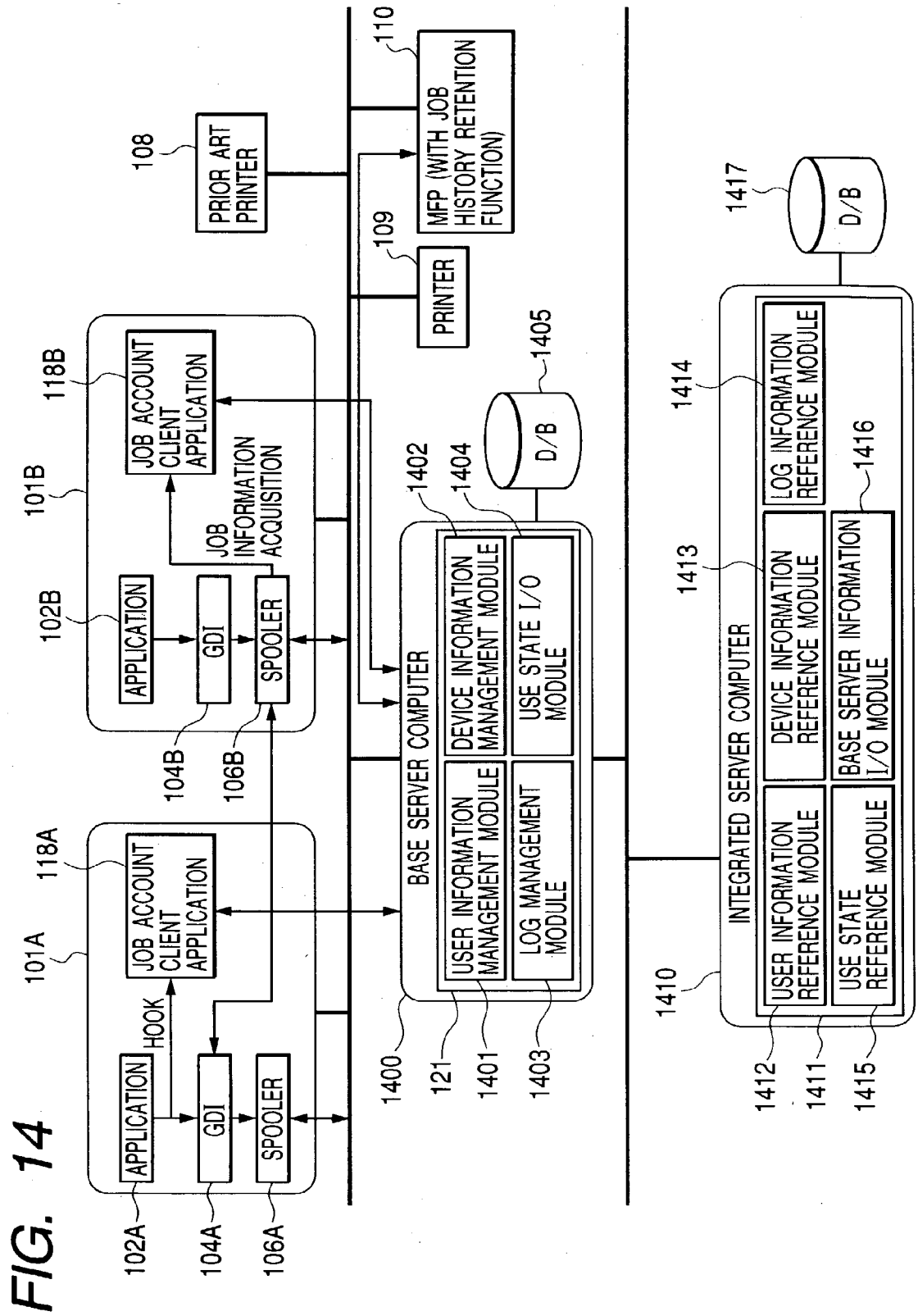

101A
118A
JOB ACCOUNT CLIENT APPLICATION
APPLICATION
HOOK
GDI
SPOOLER
102A
104A
106A
101B
118B
JOB ACCOUNT CLIENT APPLICATION
JOB INFORMATION ACQUISITION
102B
APPLICATION
GDI
SPOOLER
104B
106B
108
PRIOR ART PRINTER
110
MFP (WITH JOB HISTORY RETENTION FUNCTION)
109
PRINTER
1405
D/B
1402
1404
BASE SERVER COMPUTER
USER INFORMATION MANAGEMENT MODULE
DEVICE INFORMATION MANAGEMENT MODULE
LOG MANAGEMENT MODULE
USE STATE I/O MODULE
1400
121
1401
1403
1410
1413
1414
1417
D/B
INTEGRATED SERVER COMPUTER
USER INFORMATION REFERENCE MODULE
DEVICE INFORMATION REFERENCE MODULE
LOG INFORMATION REFERENCE MODULE
USE STATE REFERENCE MODULE
BASE SERVER INFORMATION I/O MODULE
1416
1412
1411
1415

FIG. 15

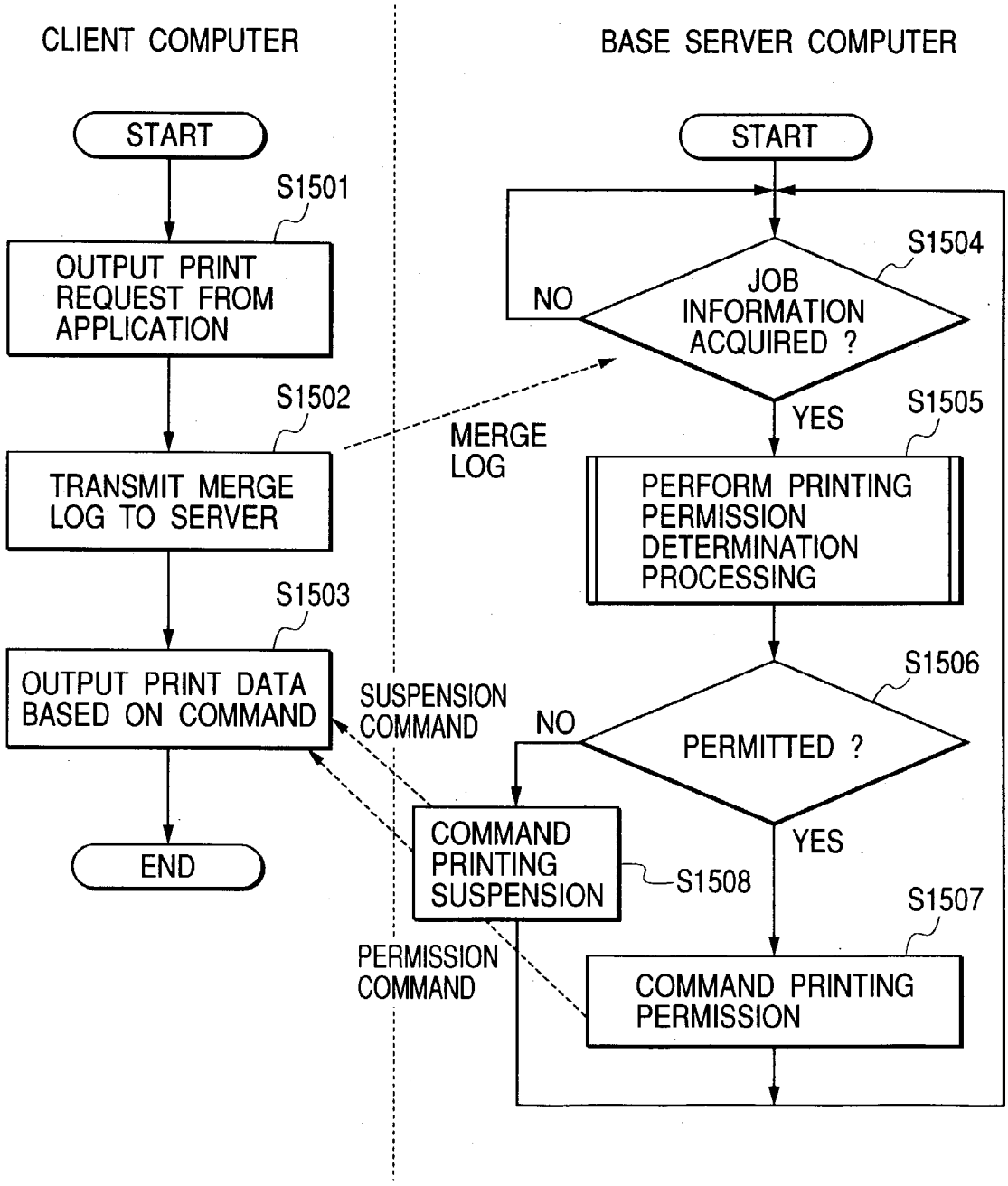

CLIENT COMPUTER
BASE SERVER COMPUTER
START
S1501
OUTPUT PRINT REQUEST FROM APPLICATION
S1502
TRANSMIT MERGE LOG TO SERVER
MERGE LOG
S1503
OUTPUT PRINT DATA BASED ON COMMAND
SUSPENSION COMMAND
END
START
S1504
JOB INFORMATION ACQUIRED ?
NO
YES
S1505
PERFORM PRINTING PERMISSION DETERMINATION PROCESSING
S1506
PERMITTED ?
NO
YES
COMMAND PRINTING SUSPENSION
S1508
S1507
PERMISSION COMMAND
COMMAND PRINTING PERMISSION

FIG. 16

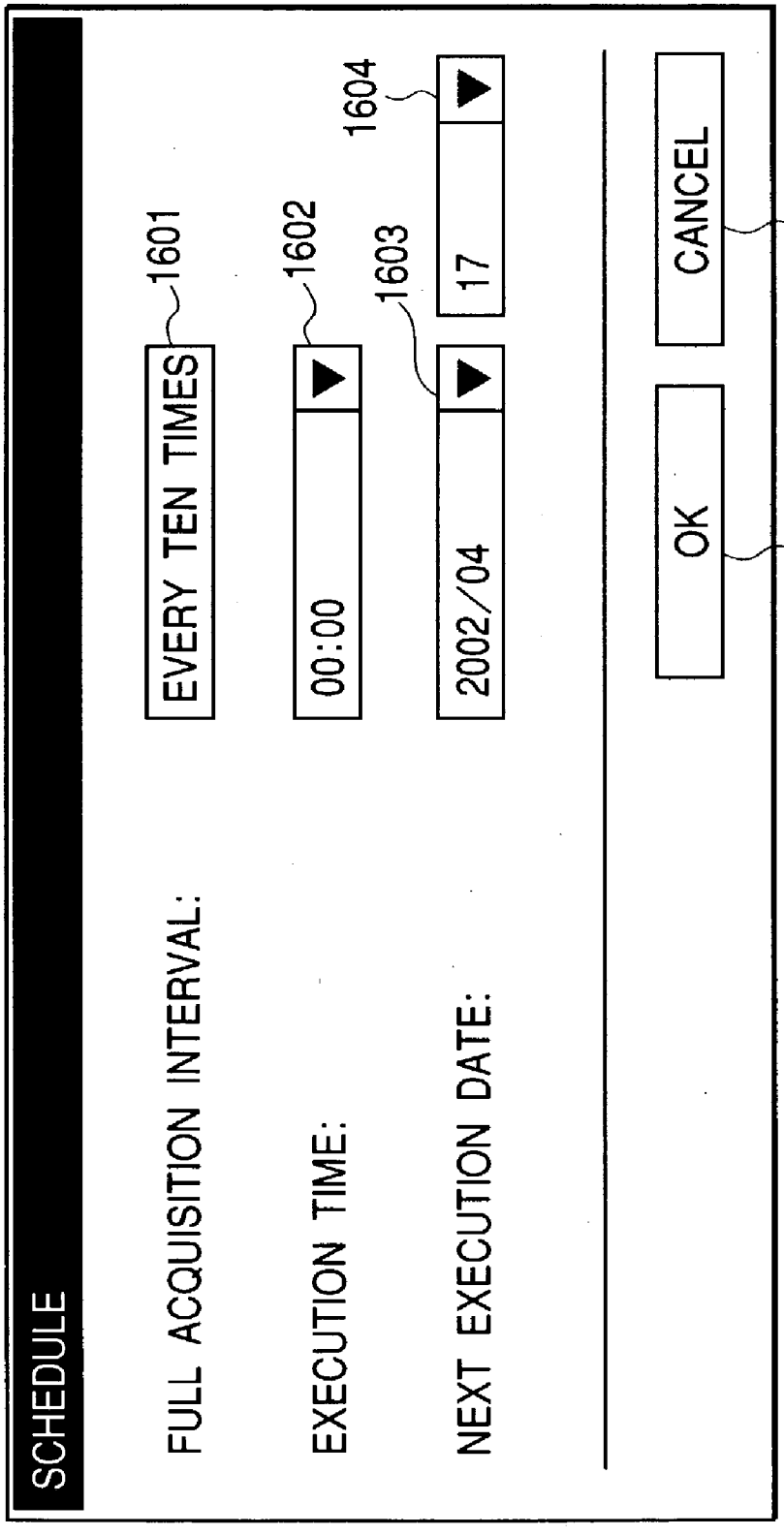

SCHEDULE
FULL ACQUISITION INTERVAL:
EVERY TEN TIMES
1601
EXECUTION TIME:
00:00
1602
NEXT EXECUTION DATE:
2002/04
1603
17
1604
OK
1605
CANCEL
1606

FIG. 17

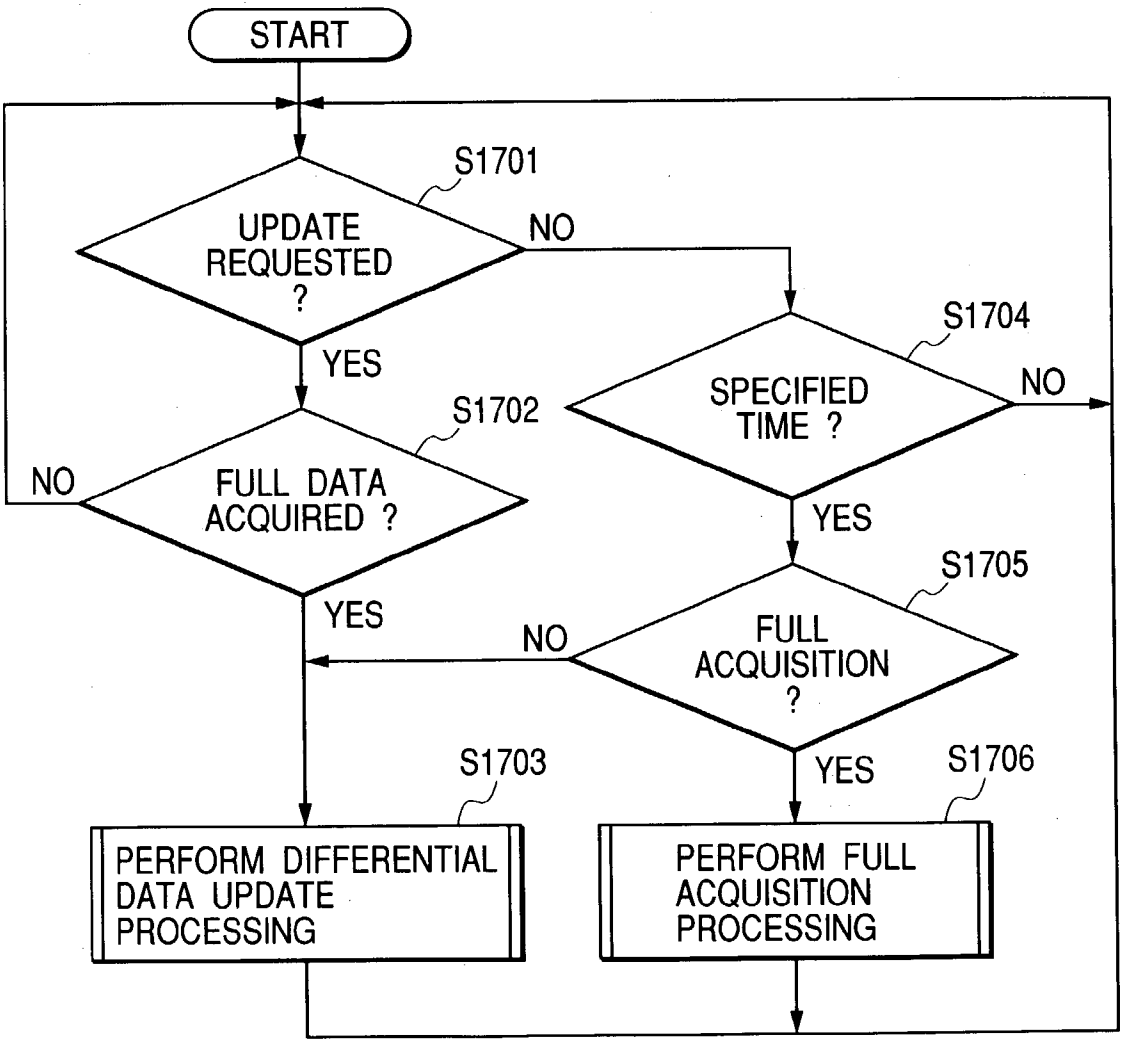

START
S1701
UPDATE REQUESTED ?
NO
YES
S1702
FULL DATA ACQUIRED ?
NO
YES
S1704
SPECIFIED TIME ?
NO
YES
S1705
FULL ACQUISITION ?
NO
YES
S1703
PERFORM DIFFERENTIAL DATA UPDATE PROCESSING
S1706
PERFORM FULL ACQUISITION PROCESSING

FIG. 18

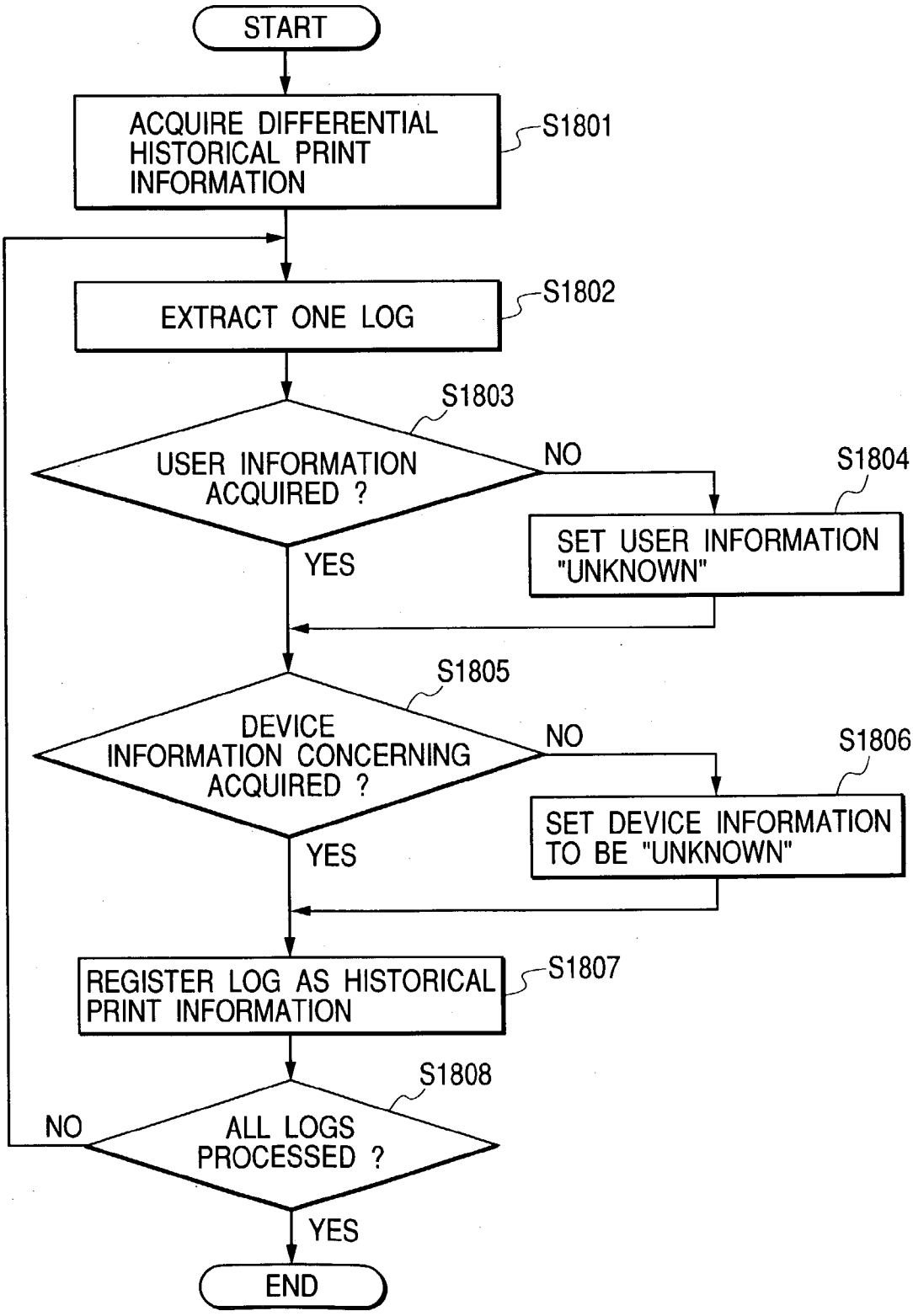

START
ACQUIRE DIFFERENTIAL HISTORICAL PRINT INFORMATION
S1801
EXTRACT ONE LOG
S1802
S1803
USER INFORMATION ACQUIRED ?
NO
S1804
SET USER INFORMATION "UNKNOWN"
YES
S1805
DEVICE INFORMATION CONCERNING ACQUIRED ?
NO
S1806
SET DEVICE INFORMATION TO BE "UNKNOWN"
YES
REGISTER LOG AS HISTORICAL PRINT INFORMATION
S1807
S1808
NO
ALL LOGS PROCESSED ?
YES
END

FIG. 19

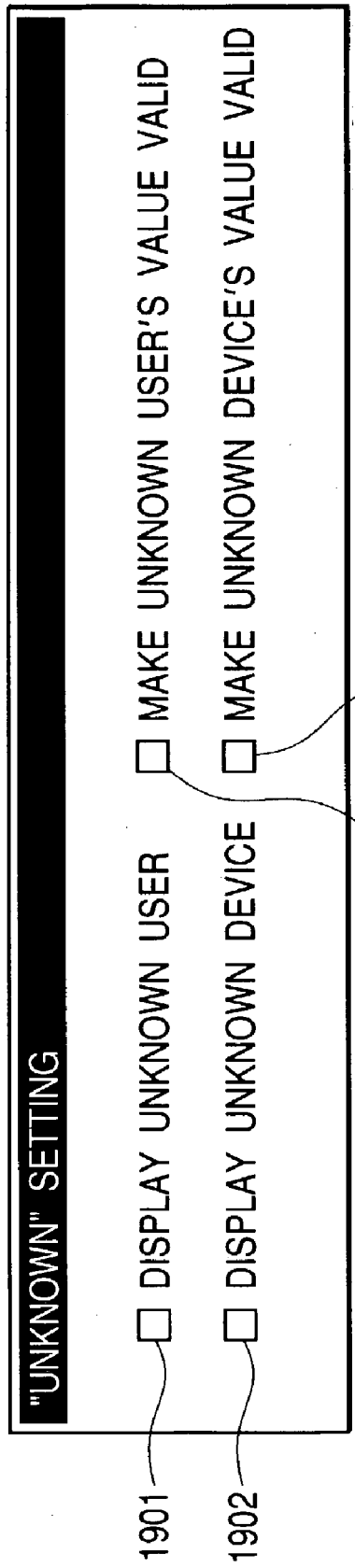
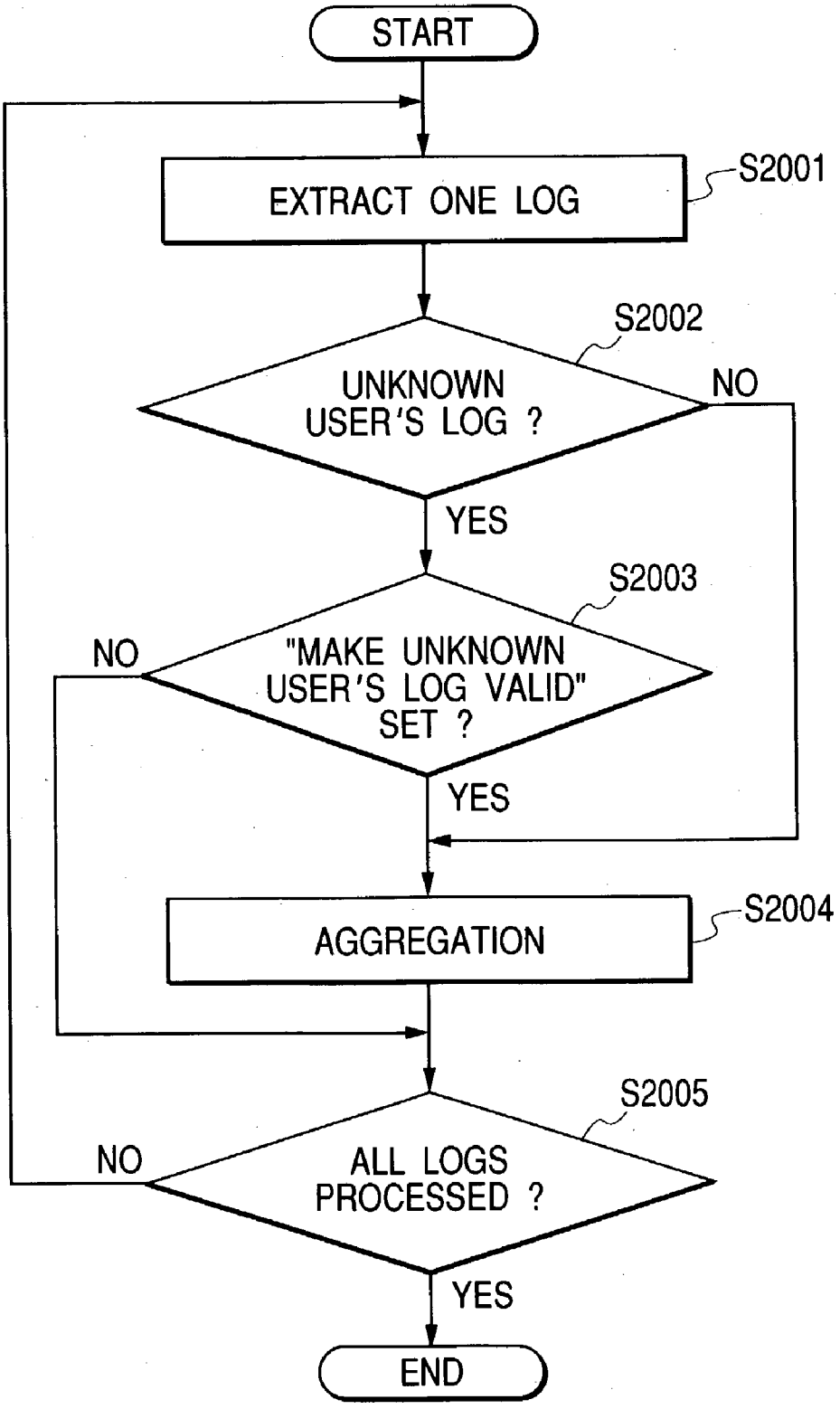

"UNKNOWN" SETTING
DISPLAY UNKNOWN USER
DISPLAY UNKNOWN DEVICE
MAKE UNKNOWN USER'S VALUE VALID
MAKE UNKNOWN DEVICE'S VALUE VALID
1901
1902
1903
1904

FIG. 20


START
EXTRACT ONE LOG
S2001
S2002
UNKNOWN USER'S LOG ?
NO
YES
S2003
"MAKE UNKNOWN USER'S LOG VALID" SET ?
NO
YES
AGGREGATION
S2004
S2005
ALL LOGS PROCESSED ?
NO
YES
END

MANAGEMENT SYSTEM, INFORMATION PROCESSING APPARATUS, AND MANAGEMENT METHOD AND PROGRAM

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to a management system of a usage amount at the time when a user uses a peripheral, which is connected to a network, such as a printer, a scanner, a copier, or a multi function peripheral (MFP) that integrate them, in a network environment constituted of a client computer, a server computer, etc.

[0003] 2. Related Background Art

[0004] Up to now, they have known information processing apparatuses that manage usage amounts or information, relating to data processing executed by use, when each user uses a peripheral such as a printer, a scanner, a copier, or a multi function peripheral (MFP) integrating them, via a network.

[0005] In such an information processing apparatus, for example, as disclosed in Japanese Patent Application Laid-Open No. 10-161823 (U.S. Pat. No. 6,202,092), one server computer manages the use of a peripheral.

SUMMARY OF THE INVENTION

[0006] Nevertheless, in conventional technology as the above, one server computer should manage plenty of peripherals when there are a lot of peripherals. In addition, in this case, peripherals that one server computer can manage are limited to ones that exist in the same building or room, or the like.

[0007] Hence, it was difficult to integrally manage a use status of peripherals in a wide range or across two or more bases by using a communication path such as the Internet. In addition, even if a plurality of server computers were installed in each base, the plurality of server computers managed only usage amounts and use statuses of peripherals under management, and information collected by each server computer were utilized individually and independently.

[0008] Hence, the present invention is to solve the tasks of conventional technology described above, and aims at making it possible to centrally manage the use of peripherals in a wide range or across a plurality of bases.

[0009] Moreover, the present invention aims at efficiently collecting information collected in a wide range or across a plurality of bases without increasing a load.

[0010] In addition, when a new user or apparatus, or the like is added in each base, a mismatch is caused between information, having already been grasped, and information collected from each base, and hence, it is not possible to correctly manage states of peripherals. On the other hand, it becomes impossible to perform accurate management in the case of unconditionally discarding the information, where the mismatch is caused, in order to avoid it, and a communication load and a processing load increase in the case of acquiring all the past information every time.

[0011] Hence, the present invention is to solve the tasks of conventional technology described above, and aims at making it possible to centrally manage the use of peripherals in a wide range or across a plurality of bases.

[0012] Moreover, the present invention aims at making it possible to perform information management corresponding to user's intention by also flexibly corresponding to the case that a new user or apparatus, or the like is added.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] **FIG. 1** is a diagram showing an example of a system that becomes a prerequisite of an embodiment of a job account system;

[0014] **FIG. 2** is a block diagram showing an example of the construction of a client computer and a server computer;

[0015] **FIG. 3** is a block diagram showing an example of the construction of a printer;

[0016] **FIG. 4** is a map showing an example of a memory map of a job account server **121**;

[0017] **FIG. 5** is a map showing constituent contents of the construction of an FD to which a job account program and related data are loaded;

[0018] **FIG. 6** is a schematic diagram showing an example of loading a job account program and related data, which are recorded in an FD, into a computer system;

[0019] **FIG. 7** is a table showing an example of a hook log;

[0020] **FIG. 8** is a table showing an example of a driver log;

[0021] **FIG. 9** is a table showing an example of an API log;

[0022] **FIG. 10** is a table showing an example of a transmission unit log;

[0023] **FIG. 11** is a table showing an example of a merge log generated by merge processing;

[0024] **FIG. 12** is a table showing an example of a device log;

[0025] **FIG. 13** is a table showing an example of a final log that a job log-merging unit **123** generates;

[0026] **FIG. 14** is a basic structural diagram of an entire management system;

[0027] **FIG. 15** is a flow chart explaining a flow of processing performed in a client computer **101** (**101**A and **101**B) and a base server computer **1400**;

[0028] **FIG. 16** is a drawing showing an example of a screen where a schedule is set so that an integrated server computer **1410** may acquire the historical print information of the base server computer **1400**;

[0029] **FIG. 17** is a flow chart showing a flow of processing at the time when the integrated server computer **1410** acquires the historical print information of the base server computer **1400**;

[0030] **FIG. 18** is a flow chart showing the processing at step **1703** in **FIG. 17**;

[0031] **FIG. 19** is a screen shot showing a setting screen for performing the setting concerning the treatment of "unknown"; and

[0032] **FIG. 20** is a flow chart showing the aggregate processing of performing aggregation.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0033] Hereafter, an example of embodiments of the present invention will be explained with referring to drawings.

[0034] **FIG. 1** is a diagram showing an example of a system that becomes a prerequisite of an embodiment of a job account system. Printers **108** to **110** that are preferable examples of an image forming apparatus of the present invention that are construction members in **FIG. 1** will be described.

[0035] Printers differently operate according to their functions. For example, the printer **108** is a most general one, and prints data received from a transmission unit in a client computer. In addition, the data processing that a device is required is called a job, and in particular, the printing processing based on print data is called a printing job. The printer **109** sends a total number of delivery pages of the job at the time of job end to the transmission unit in the client computer **101**. After printing, the printer **110** stores job information in memory in the printer.

[0036] The client computer **101** in **FIG. 1** performs printing as follows. An application **102** operates a GDI (a subsystem processing graphic drawing) **104** of an OS by calling an application programming interface (API). A printer driver **105** converts a request from the GDI **104** into a form suitable to a printer performing printing to transmit the request to a spooler **106**.

[0037] In the client computer **101**, a job account client **118** hooks (monitors and recognizes) a GDI call of the application **102**. The job account client **118** acquires a number of delivery sheets or a number of pages of a job that the application issued by counting the API performing page breaks or paper ejections. The job information acquired owing to this is called a hook log.

[0038] In addition, the job account client **118** can receive job information from a printer driver in the client computer by beforehand defining a protocol with and communicating with the printer driver. The job information acquired owing to this is called a driver log.

[0039] Furthermore, the job account client **118** acquires a number of delivery sheets or a number of pages of a job by monitoring a spooler periodically, and acquiring job information in the API when there is the job to be spooled. The job information acquired owing to this is called an API log.

[0040] Furthermore, the job account client **118** acquires a total number of delivery pages, which the printer returns, by defining a protocol with and communicating with the transmission unit. The job information acquired owing to this is called a transmission unit log.

[0041] Moreover, the job account server acquires job information from a printer by defining a protocol with and communicating with a printer in the case of the printer, storing the job information in the printer, such as a printer **110** that is a preferable example of the image forming apparatus. The job information acquired owing to this is called a device log. Though the device log is stored in RAM or an HDD in the printer, due to the limitation of memory capacity, it frequently happens that it is not possible to store all of the information such as a document name transmitted from the transmission unit. In addition, the image forming apparatus includes a facsimile, a copier, a copying system combining a printer, which is a kind of copier, and a scanner, and a compound machine formed of these besides a printer.

[0042] The job account client **118** provides the priority of which is to be selected from among the hook log, driver log, API log, transmission unit log, and device log, and transmits any one to the job account server.

[0043] A job account server **121** selects a job log according to a reading requester's request to perform aggregate processing such as aggregation by unit time, by device, and by job issuer.

[0044] **FIG. 1** will be explained in further detail.

[0045] The client computer **101** composed of a standard PC or a work station transmits print data to either of the printers **107**,**108**, and **109** through the transmission unit **107** when printing is instructed from the application **102**. The job account client **118** that is a software unit executed by a CPU in the client computer collects logs concerning various types of printing processing to transmit them to the job account server **121** through a job log transmission unit **117**. At this time, the job log transmission unit that is a part of the job account client can also transmit the logs to a job log receiving unit by controlling a transmission part that the OS provides.

[0046] When a printing instruction is issued to the application **102** in the client computer **101**, the application **102** calls the GDI for drawing. A hook unit **103** is built in by the job account client **117**, and stores information such as which GDI call is called by any parameter, and a called frequency by hooking the GDI calls. The history established on the basis of this information is called a hook log **111**.

[0047] An example of the hook log **111** is shown in **FIG. 7**. The hook unit **103** may change a document name on the basis of a specific criterion according to circumstances. The document name and this modification processing will be described later.

[0048] The printer driver **105** converts the GDI call into print data that the printer can interpret according to print settings to transmit the print data to the spooler **106**, and extracts printed information from the contents of the conversion to transmit the printed information to the job account client **117** as a driver log **112**.

[0049] The job account client **117** monitors whether a job stays in the spooler **106**. If a job stays, the job account client **117** issues an API specified by the OS to acquire the printed information, and acquires the printed information as an API log **113**.

[0050] The transmission unit **107** communicates with the printer designated among the printers **108**,**109**, and **110** from the application via a network to transmit print data if the printer is in a ready state. At this time, for instance, if a command in Printer Job Language of Hewlett-Packard Company that is described above is used, it is possible to acquire a number of pages that the job concerned delivered in the printer **109** after completing the delivery of all the pages of the printing job. A transmission unit (port monitor) **106**

sends the job information such as the number of pages received from the printer **109** to the job account client **110**. The printing log created on the basis of this information is called a transmission unit log **114**.

[**0051**] In this manner, in the job account client (acquisition means) **110**, four logs, that is, the hook log, driver log, API log, and transmission unit log are acquired, and created for one printing job. Nevertheless, four logs can not be necessarily taken at any time. For example, if the application **102** sends print data directly to the spooler **106**, it is not possible to acquire the hook log **111** and driver log **112**. If a protocol is not specified between the printer driver **105** and job account client **117**, it is not possible to acquire the driver log **112**. If there is no function of notifying the transmission unit of a number of delivery pages like the printer **108**, it is not possible to acquire delivery page information in the transmission unit log **114**. Nevertheless, by acquiring logs in a plurality of processing relating to the job, a probability of being able to acquire the log rises totally even if log acquisition is impossible in a part of processing.

[**0052**] The printer **108** interprets the job sent from the transmission unit **107**, and executes printing. The printer **109** interprets the job sent from the transmission unit **107**, and executes printing. After the end of the printing, the number of delivery pages and the number of delivery sheets are replied to the transmission unit **107**. The printer **110** interprets the job sent from the transmission unit **107**, and executes printing. After of the end of the printing, the job information is stored in an HDD (described later) **304**.

[**0053**] A job log merging unit **116** merges the hook log, driver log, API log, and transmission unit log to send its result to the job log transmission unit **117**. Merge processing will be described later. The job log transmission unit **117** transmits a job log to a job log receiving unit **122** of the job account server **121**.

[**0054**] **FIG. 2** is a block diagram showing an example of the construction of a client computer and a server computer constituting the job account system shown in **FIG. 1**. In **FIG. 2, a** CPU **201** is a central processing unit, and performs the control of this entire computer apparatus, arithmetic processing, etc. ROM **202** is read only memory, and is a storage area of information of the system boot program. RAM **203** is random-access memory, and is a data storage area without limitation of use, the RAM **203** which is an area where software units, that is, the operating system, application **102**, hook unit **103**, GDI **104**, printer driver **105**, transmission unit **107**, and job account client **118**, and, a program for communication control etc. are loaded from the HD unit **205** and ROM **202**, and are executed by the CPU **201**. A KBC **204** is a keyboard control unit, and receives input data from a keyboard to transmit the input data to the CPU. A CRT **205** is a display control unit (display control means), and controls display in a display apparatus. An external storage apparatus **206** is an FD, which is a floppy (R) disk drive, an HD, which is a hard disk drive, or SRAM, which is nonvolatile storage, and stores a program and data that are referred to or loaded to the RAM if necessary at an execution time. A communication unit **207** controls network communication, and can communicate with another computer and a peripheral, which are connected to a network as explained in **FIG. 1**, and is controlled by the transmission unit **107** and job log transmission unit **117**. A system bus **208** becomes a data path between the above-described components.

[**0055**] In principle, the job account server **121** can also have the similar construction in view of hardware. The HDD of the job account server **121** stores a job log merging unit **123**, a job log recording unit **124**, a job log receiving unit **122**, and a job log aggregating unit **125**. In addition, the job log receiving unit **122** controls the communication unit **207**.

[**0056**] **FIG. 3** is a block diagram showing an example of construction of a printer constituting the job account system shown in **FIG. 1**. In **FIG. 3, a** controller unit **305** controls the entire printer. A CPU **301** is a central processing unit constituting the controller, and performs the control of this controller, arithmetic processing, etc. ROM **302** is read only memory, and is a storage area of information etc. of the system boot program. RAM **303** is random-access memory, and is a data storage area without limitation of use, and nonvolatile storage **304** is a HD, which is a hard disk, SRAM, or the like.

[**0057**] The RAM **303** is an area where the operating system, and programs for communication control, engine controller, etc. are loaded and executed. An engine unit **306** of this apparatus performs printing operation and image reading under the control of the controller. A communication unit **307** controls network communication, and can communicate with the client computer and server computer as explained in **FIG. 1**. A user operation I/O unit **308** performs interactive setting work with a user. A system bus **309** becomes a data path between the above-described components.

[**0058**] **FIG. 4** is a map showing an example of a memory map of the job account server **121** shown in **FIG. 1**. Reference numeral **41** denotes a basic I/O program. Reference numeral **42** denotes a part of the memory map where the operating system like a window system etc. is loaded into the RAM **203** and becomes executable. Reference numeral **43** denotes a memory map in a state where a program of the present invention is loaded into the RAM **203** and became executable. Reference numeral **44** is a memory map in a state where related data is loaded into the RAM **203** and became executable. Reference numeral **45** shows a memory map of the work memory that each program uses.

[**0059**] **FIG. 5** is a map showing constituent contents of the construction of an FD to which the job account program and related data that are examples of an embodiment of the present invention are loaded. The job account program and related data that are examples of an embodiment of the present invention are recorded in a floppy (R) disk (FD). Volume information **51**, directory information **52**, an application execution file **53**, and a data file **54** related to the job account program are recorded in a recording area **50** of the floppy (R) disk (FD). In addition, it is needless to say that the recording area **50** can be a DVD or an HDD (not shown) in an external server apparatus network-connected to a computer **62**, and the job account program of the present invention (each unit in the client computer and each unit in the job account server **121** that are shown in **FIG. 1**) can be also implemented by being downloaded via the network. The computer **62** is equivalent to the computer shown in **FIG. 2**.

[**0060**] **FIG. 6** is a schematic diagram showing an example of loading a job account program and related data, which are

examples of an embodiment of the present invention and are recorded in an FD, into a computer system. It is possible to load the job account program and related data, which are recorded in the floppy (R) disk (FD) **60**, through an FD drive (DKC) **61** into the computer **62**. When the floppy (R) disk (FD) **60** is set in the FD drive (DKC) **61**, program units and related data for achieving this job account program are read from the floppy (R) disk (FD) **60** and are loaded into the RAM **203**, under the control of the operating system **42** and basic I/O program **41**, and become available. The computer **62** can be used to achieve the client computer **101** and job account server **121** that are described in **FIG. 1**. In addition, the "program unit etc. for achieving the job account program" correspond to each software unit in the client computer **101** or in the job account server **121** that are shown in **FIG. 1**. Of course, it is needless to say that a storage medium is not limited to the FD, but includes a DVD, a CD-ROM, or the like.

[**0061**] **FIG. 7** is a table showing an example of a hook log in an embodiment of the present invention. In the hook log, it is possible to acquire a job ID (job identifier), an application name, a number of logical pages, and a document name. The job ID is acquired from the GDI when printing start is instructed to the GDI, and can uniquely designate a printing job in the client computer **101**.

[**0062**] **FIG. 8** is a table showing an example of a driver log in this embodiment. The driver log is historical information that the job account client can acquire from the driver at a printing execution time. It is possible to acquire a job ID that is an identifier of a job relating to print execution, paper size of print data, N-UP information, information on a number of physical pages, and duplex/simplex information that the job includes, from the driver log. The paper size, N-UP information, number of physical pages and duplex/simplex information are acquired from the print setting that is set in the printer driver at printing or by a default input. N-UP is the setting of how many logical pages are allocated in a form. The number of physical pages has the unit that a simplex of a form outputted at printing is counted as one.

[**0063**] For example, when 2-UP printing processing is performed in duplex printing at the time of four logical pages, physical pages become 2. Then, the physical pages are printed on both sides of the form, and one sheet of form is ejected. A first logical page and a second logical page are printed on a top, and a third logical page and a fourth logical page are printed on a back.

[**0064**] **FIG. 9** is a table showing an example of the API log in this embodiment. In the API log, it is possible to acquire a job ID, an owner's name, and spool data size.

[**0065**] **FIG. 10** is a table showing an example of the transmission unit log in an embodiment of the present invention. In the monitor log, it is possible to acquire a job ID. When transmission is performed to the printer **108**, it is possible to acquire a number of delivery sheets etc. in addition to this.

[**0066**] In each of the hook log, driver log, API log, and transmission unit log, an area where flag information showing whether the job concerned ends the processing and information showing a log type are recorded is provided.

[**0067**] Here, the operation of the job log merging unit **116** in the client computer will be explained. The job log merging unit **116** receives each log, and records its contents in an HDD etc. Then, the job log merging unit **116** starts periodically in a predetermined period, and performs the merge processing of logs per job.

[**0068**] Though logs are merged-in this embodiment by making a job ID a key, it is also good to merge logs by making a document data name, which will be described later, or a new document data name a key.

[**0069**] **FIG. 11** is a table showing an example of a merge log generated by merge processing. This merge log provides items of a hook log, a driver log, an API log, and a transmission unit log as shown in the table.

[**0070**] Next, the operation of the job account server **121** will be explained. A job log receiving unit **122** that is a preferable example of acquisition means of the present invention acquires and stores a job sent from the job log transmission unit **117**. In addition, the job log receiving unit **122** performs the polling of the printer-**3110** periodically, and acquires job information if there is the job information being not acquired yet. This job information is made a device log. Though the device log is stored in the RAM **303** or HDD **304**, it is not desirable to store all the information such as a document name, transmitted from the transmission unit **107**, because of the limitation of memory capacity. Here, it is assumed that the upper limit of the document name is eight characters. **FIG. 12** is a table showing an example of the device log.

[**0071**] A device job log merging unit **123** that is a preferable example of managing means of the present invention merges a merge log **127**, sent from the job account client, and the device log **115**. A job log recording unit **124** receives the job log from the job log merging unit **123** and stores it in the HD **206** etc. A job log aggregating unit **125** performs aggregate processing such as finding of a number of printed pages per unit period, or a number of printed pages per owner by a reader's command.

[**0072**] The job log merging unit **123** retrieves and determines whether there is an unprocessed device log where an item of the document name is the same as that of the merge log, and substitutes or adds the number of delivery pages and number of delivery sheets in the device log for/to those in the merge log if it is determined to be present.

[**0073**] **FIG. 13** is a table showing an example of a final log that the job log merging unit **123** generates. The job log merging unit **123** transmits the final log to a job log recording unit **124**.

[**0074**] **FIG. 14** is a basic structural diagram of an entire management system as this embodiment. Reference numerals **101**A and **101**B denote client computers. Reference numeral **1400** denotes a base server computer. Reference numeral **1410** denotes an integrated server computer.

[**0075**] Reference numerals **108** to **110** denote printers. Nevertheless, it is assumed that the printer **110** is a multi function periphery (MFP) that has a function (printing history retaining function) of retaining printing history, and printers **108** and **109** do not have the printing history retaining function. The MFP **110** comprises a printer function, a copy function, a scanner function, and the printing history retaining function.

[0076] The computers **101**A, **101**B, **1400**, and **1410**, and the printers **109** and **108**, and MFP **110** can respectively communicate with other apparatuses via a network.

[0077] In the client computer **101**A, when the application **102**A calls the GDI (Graphics Device Interface) **104**A, the GDI **104**A generates print data according to a command from the application **102**A to transmit print data to the spooler **106**A.

[0078] The job account client application **118**A monitors (hooks) the application **102**A calling the GDI **104**A to store a hook log relating to the print data.

[0079] The spooler **106**A communicates with a peripheral via the network, and if the peripheral is ready, the print data is transmitted. In addition, when the client computer **101**B functions as a printer server, the print data that the GDI **104**A generated is transmitted to the spooler **106**B.

[0080] The job account client application **118**A transmits the merge log, periodically stored, to the job account server application **121**.

[0081] In the client computer **101**B, the application **102**B, GDI **104**B, and spooler **106**B operate similarly to the application **101**A, GDI **104**A, and spooler **106**A, respectively.

[0082] A job account client application **118**B monitors the spooler **106**B periodically. If there is a printing job spooled in the spooler **106**B, the job account client application **118**B acquires information on the printing job by using an API (Application Program Interface). Specifically, the job account client application **118**B acquires a number of delivery sheets and a number of pages of the printing job. Then, the job account client application **118**B transmits the merge log, periodically stored, to the job account server application **121**.

[0083] If the O/S comprises a function of sending a status change of the spooler **106**B to another application, it is acceptable that the job account client application **118**B does not monitor the spooler **106**B periodically. The OS only has to send to the job account client application **118**B that the printing job is spooled in the spooler **106**B.

[0084] In addition, reference numerals **102**A and **102**B are equal to reference numeral **102** in **FIG. 1**, reference numerals **104**A and **104**B are equal to reference numeral **104** in **FIG. 1**, reference numerals **106**A and **106**B are equal to reference numeral **106** in **FIG. 1**, and reference numerals **118**A and **118**B are equal to reference numeral **118** in **FIG. 1**.

[0085] In the server computer **1400**, the job account server application **121** includes a user information management unit **1401**, a device information management unit **1402**, a log management unit **1403**, and a use state I/O unit **1404**. Among these, the user information management unit **1401** stores and manages user information in memory storage **1405**, not shown, in server computer **1400**.

[0086] Furthermore, the device information management unit **1402** investigates whether a peripheral connected to the network has a device log retaining function, and stores information on a device, having the device log retaining function, and other device information in the memory storage **1405** in the server computer **1400** and manages them.

[0087] The user information includes a computer log-in name (a name inputted so that a user may log in the computer) and a network log-in name (a name inputted so that the user may log in the network system) of the user using a peripheral, and is information for specifying (identifying) the user using the peripheral. In addition, the device information includes printing speed, color information, a network address, etc. of the peripheral.

[0088] The log management unit **1403** receives the merge log **127** from the job account client applications **118**A and **118**B, and also stores the merge log in the memory storage **1405**. In addition, the log management unit **1403** requests a device log to a peripheral (for example, the printer **1140**) having a function of retaining the device log **115**, and acquires the device log through the use state I/O unit **1404**. This processing may be periodically performed. Furthermore, it is also acceptable that, when the memory capacity of a device log storage unit (not shown) of a peripheral decreases, it is reported to the use state I/O unit, and at that time, this processing is performed.

[0089] The use state I/O unit **1404** acquires a use state including a device log from a peripheral having a device log retaining function. In addition, the use state I/O unit **1404** can also output a use state report (a report showing how many jobs were executed per user) to a printer and/or a monitor of a peripheral.

[0090] A supervisor can make the use state I/O unit **1404** output a use state to acquire a use state of the peripheral. The use state I/O unit **1404** outputs various use states on the basis of user information, device information, and historical print information (job information formed by summarizing a plurality of final logs in **FIG. 13**) which are stored in the memory storage **1405**. The supervisor designates an output form (output format) for them to the use state I/O unit. In addition, this use state output is displayed in, for example, a display unit (not shown) of the server computer **1400**. Furthermore, output data relating to the use state output may be transmitted to the client computer **101**A or **101**B, and may be displayed in a display unit (not shown) of the client computer.

[0091] The use state output outputs a usage amount (a number of sheets of forms and an amount of toner used for outputs) and the used time of a peripheral per user. In addition, a use state is separately outputted per operation mode (function) of a peripheral such as color printing or duplex printing. Furthermore, a use state is outputted per size or paper type used.

[0092] Moreover, a usage charge per peripheral or per user is outputted when a usage rate of the peripheral is set in the device information. In addition, the use state output outputs a paper type, an operation mode, a peripheral name, a printing amount, and a usage charge per user when usage rates per operation mode, per paper size, and per paper type are set.

[0093] Furthermore, a supervisor can also set use limitation per function, per user, per user department, and per used device through the user information management unit **1401** and device information management unit **1402**. There are items such as a usage rate, a number of pages, and a number of sheets of paper as the limited information that can be set. When a user exceeds the use limitation, the use state I/O unit

1404 outputs so by monitoring the historical print information and collating the historical print information with the set limitation (limitation information). It is no problem whether this is a display output on a screen or a communication output such as e-mail.

[**0094**] In addition, when a certain user's use exceeds the limitation, the log management unit **1403** makes the client computer **101**A or the peripheral itself refuse the user's use by monitoring the historical print information and collating the historical print information with the limitation set per user. Alternatively, it is also acceptable to sound an alarm by the display on a screen or by other communication means such as e-mail to the user.

[**0095**] The integrated server computer **1410** and base server computer **1400** can communicate with each other via a network such as a LAN or the Internet. In addition, it is no problem whether a number of the base server computers **1400** that communicate with one integrated server computer is single or plural.

[**0096**] In the integrated server computer **1410**, the job account integration server application **1411** includes a user information reference unit **1412**, a device information reference unit **1413**, a log reference unit **1414**, a use state reference unit **1415**, and a base server information I/O unit **1416**.

[**0097**] Among these, the user information reference unit **1412** can refer to user information, stored in the memory storage **1417**, with the base server information I/O unit **1416**.

[**0098**] In addition, it is needless to say that it is no problem whether the memory storage **1417** is installed in the integrated server computer **1410** and is managed or is installed in another computer, which can communicate with the integrated server computer **1410**, and is managed.

[**0099**] Furthermore, the device information reference unit **1413** can refer to device information, stored in the memory storage **1417**, with the base server information I/O unit **1416**.

[**0100**] The log reference unit **1414** can refer to historical print information, stored in the memory storage **1417**, with the base server information I/O unit **1416**.

[**0101**] The use state reference unit **1415** can refer to information on a device use state that is stored in the memory storage **1417**.

[**0102**] The base server information I/O unit **1416** acquires the user information, device information, and historical print information from the base server computer **1400**. In addition, the base server information I/O unit **1416** can also output a use state report of the base server computer **1400** (a report showing how many jobs were executed per user per base server computer **1400**) to a printer and/or a monitor of a peripheral.

[**0103**] A supervisor can make the base server information I/O unit **1416** output a use state to acquire a use state of the peripheral. The base server information I/O unit **1416** outputs various use states on the basis of base server information, user information, device information, and historical print information that are stored in the memory storage **1417**.

[**0104**] The supervisor designates an output form (output format) for them to the base server information I/O unit **1416**. In addition, this use state output is displayed in, for example, a display unit (not shown) of the integrated server computer **1410**. Furthermore, output data relating to the use state output may be transmitted to the client computer **101**A, and may be displayed in a display unit (not shown) of the client computer.

[**0105**] The use state output outputs a usage amount (a number of sheets of forms and an amount of toner used for outputs) and the used time of a peripheral per user registered in each base server computer **1400**. Moreover, a use state is separately outputted per operation mode (function) of a peripheral such as color printing or duplex printing. In addition, a use state is outputted per size or paper type used.

[**0106**] Furthermore, a usage charge per peripheral or per user is outputted when a usage rate of the peripheral is set in the device information. Moreover, the use state output outputs a paper type, an operation mode, a peripheral name, a printing amount, and a usage charge per user when usage rates per operation mode, per paper size, and per paper type are set.

[**0107**] **FIG. 15** is a flow chart explaining a flow of processing performed in the client computer **101** (**101**A and **101**B) and base server computer **1400**.

[**0108**] First of all, at step **S1501**, when a user instructs printing in the application **102** in the client computer **101**, the application **102** edits a document, an image, etc. and outputs a print request according to the user's instruction.

[**0109**] Next, at step **S1502**, the job account client application **1014** acquires a hook log, a driver log, an API log, and a transmission unit log, and generates a merge log from them to transmit the merge log to a base server computer.

[**0110**] Subsequently, at step **S1503**, print data is outputted to a printer **1130** through the spooler **106** on the basis of an instruction from the job account server application.

[**0111**] On the other hand, in the base server computer **1400**, when the log management unit **1403** acquires the merge log at step **S1504**, the process proceeds to step **S1505**, and the determination processing of printing permission is performed.

[**0112**] Next, the process proceeds to step **S1507** when the base server computer **1400** permits the printing at step **S1506**, and the base server computer **1400** instructs the printing permission. If the base server computer **1400** does not permit the printing, the process proceeds to step **S1508**, and the base server computer **1400** transmits an instruction of printing suspension to the printer or client computer.

[**0113**] Subsequently, the processing at the time when the integrated server computer **1410** acquires various types of information from the base server computer **1400** will be explained by using **FIGS. 11 and 12**.

[**0114**] **FIG. 16** is a drawing showing an example of a screen where a schedule is set so that the integrated server computer **1410** may acquire the historical print information of the base server computer **1400**.

[**0115**] On this screen, it is assumed that the integrated server computer **1410** acquires the historical print information of the base server computer **1400** at a fixed time every

day. In addition, if the processing of acquiring all the historical print information are performed at the same timing when a lot of base server computers **1400** exist, a load to the integrated server computer **1410** becomes large. Hence, it is possible to perform settings different from each other for respective base server computers **1400**.

[**0116**] On this screen, there are an interval input column **1601** for designating an interval of full acquisition of historical print information, a time input column **1602** for designating the execution time of processing of acquiring full or differential historical print information, a month input column **1603** and a day input column **1604** for designating data and time when the next full data acquisition will be performed, an OK button **1605** for finally establish information inputted to those columns, and a cancel button **1606** for exiting from this diagram without finally establishing those information.

[**0117**] When it becomes time inputted in the time input column **1602**, the integrated server computer **1410** acquires historical print information by giving a request to the base server computer **1400**. At this time, the integrated server computer **1410** usually acquires only the historical print information that is newly added to be increased, and acquires all the historical print information, which the base server computer **1400** has, in each interval inputted in the interval input column **1601**. The former is called differential data update processing, and the latter is called full data acquisition processing.

[**0118**] In addition, in the full data acquisition processing, all the historical print information of an object base is acquired after all the historical print information that was acquired has been already deleted. On the other hand, in the differential data update processing, only the historical print information added in the base server computer after having been acquired last time is acquired. Hence, if a department, a user, a device, or the like is added in the base server computer after the full data acquisition processing, the additional one is determined as an unknown one in the difference data update processing.

[**0119**] Hence, the full data acquisition processing is performed so as to manage the added department, user, device, or the like. In addition, it is possible to avoid a mismatch of historical print information by performing the differential data update processing without fail after the full data acquisition processing.

[**0120**] A reason why a next execution date when the full data acquisition processing will be performed is specifiable in the month input column **1603** and day input column **1604** is to make it possible to distribute a load to the integrated server computer **1410** by shifting a day when historical print information is acquired if a lot of base server computers **1400** exist. In particular, in the full data acquisition processing, since there is a possibility that a large amount of historical print information is transferred, it is remarkably effective to make it possible to designate an execution date in each base server computer.

[**0121**] **FIG. 17** is a flow chart showing a flow of processing at the time when the integrated server computer **1410** acquires the historical print information of the base server computer **1400**.

[**0122**] First of all, at step **S1701**, the integrated server computer **1410** determines whether a supervisor instructed an update request of the differential data of the base server computer **1400** in the job account integration server application **1071**.

[**0123**] When instructed, it is determined whether the full data acquisition processing was performed before at step **S1702**. If not instructed, differential historical print information is acquired from the base server computer **1400** at step **S1703**. In addition, the processing at step **1703** is called differential data update processing.

[**0124**] The process returns to step **S1701** when the full data acquisition was performed before.

[**0125**] When the update request of differential data was not instructed at step **S1701**, it is determined at step **S1704** whether it is the time that the periodic update processing is designated.

[**0126**] The periodic update processing means the update processing of historical print information performed at the time set on the schedule setting screen in **FIG. 16**. The process returns to step **S1701** if it is not the designated time. When it becomes the designated time, the integrated server computer **1410** confirms set contents, and confirms whether the full data acquisition processing is performed or the differential data update processing is performed.

[**0127**] When performing the differential data update processing, the integrated server computer **1410** acquires the differential historical print information from the base server computer **1400** at step **S1703**. When performing the full data acquisition processing, the integrated server computer **1410** acquires the full historical print information from the base server computer **1400** at step **S1706**.

[**0128**] **FIG. 18** is a flow chart showing the processing at step **1703** in **FIG. 17**. Differential historical print information is acquired from the base server computer **1400** at step **S1801**. Then, one log is extracted from the acquired historical print information at step **S1802**. For example, the extracted log has the contents in **FIG. 13**.

[**0129**] Next, an item concerning a user (owner's name) in the extracted log is examined at step **S1803**, it is determined whether user information has already been acquired by the integrated server computer. The process moves to step **S1805** when already registered. When not registered yet, it is determined that a user that the integrated server computer does not know is added, the process proceeds to step **S1804**, and the data of the item concerning the user (owner's name) in the extracted log is replaced with "unknown (unknown user)".

[**0130**] Subsequently, an item concerning a device (device name) in the extracted log is examined at step **S1805**, it is determined whether device information has already been acquired by the integrated server computer. The process moves to step **S1807** when already registered. When not registered yet, it is determined that a device that the integrated server computer does not know is added, the process proceeds to step **S1806**, and the data of the item concerning the device (device name) in the extracted log is replaced with "unknown (unknown device)".

[**0131**] The log is registered in the memory storage **1417** as historical print information at step **S1807**. Then, it is determined at step **S1808** whether the processing at steps **S1802** to **S1806** was performed for all logs, and the processing is ended when performed.

[0132] Though the items concerning a user and a device are confirmed in the above-described example, it is also possible to perform similar processing on items concerning paper type, card information, etc. besides them.

[0133] Usually, the information on a user and a device is registered in the base server computer to be reported from the base server computer to the integrated server computer, or is registered in the integrated server computer. Then, these types of information are collated with historical print information, and a number of printed sheets per user, a number of printed sheets per department where a user belongs, and a number of printed sheets per device are aggregated.

[0134] As for whether the information on a user or a device is registered, if the user or device that is not included in the historical print information acquired in the above-described data acquisition processing is included in the historical print information, acquired in the differential update data processing, when the differential data update processing is performed after the full data acquisition processing, it is possible to determine this as "unknown".

[0135] In addition, "unknown" may be one type in the integrated server computer, or it is also good to classify the "unknown" into "unknown in base A" every base.

[0136] **FIG. 19** is a screen showing a setting screen for performing the setting concerning the treatment of "unknown".

[0137] First of all, reference numeral **1901** is for setting whether an unknown user is displayed as "unknown". Reference numeral **1902** is for setting whether an unknown device is displayed as "unknown".

[0138] In addition, reference numeral **1903** is for setting whether the historical print information of an unknown user is made valid. Reference numeral **1904** is for setting whether the historical print information of an unknown device is made valid;

[0139] Besides these, it is also possible to set whether the historical print information of an unknown user and the like that is added in the above-described processing is included in the aggregation work that a supervisor performs.

[0140] Furthermore, it is also possible to grip a flow of job acquisition every base by aggregating what is made unknown in each base.

[0141] **FIG. 20** is a flow chart showing the aggregate processing of performing aggregation on the basis of these settings. When a supervisor executes aggregate processing by designating a desired aggregation condition, the following processing is performed.

[0142] First of all, one log that agrees with the aggregation condition is extracted at step **S2001**. Then, it is determined at step **S2002** whether the extracted log is a log whose user is unknown.

[0143] When the log is a log whose user is not unknown, the process proceeds to step **S2004**, and aggregate processing is performed so as to reflect the contents of the log concerned in aggregation result.

[0144] In the case of the log whose user is unknown, it is determined at step **S2003** on the basis of the contents of the setting in **FIG. 19** whether "make unknown user's log valid" is designated. If not designated, the process proceeds to step **S2005** so as not to reflect the log concerned in the aggregation result. If "make unknown user's log valid" is designated, the process proceeds to step **S2004** so as to perform aggregation by including the log concerned.

[0145] Finally, it is confirmed at step **S2005** whether processing has been performed for all logs, and the above-described steps are repeated until processing is fully performed.

[0146] Respective computers execute processing shown in flow charts shown in respective charts (**FIGS. 15, 17, 18, 20**, etc.) in this embodiment, and, respective units shown in **FIGS. 1 and 14** on the basis of programs. Then, also in the case that an information group including a program is supplied by a storage medium such as a CD-ROM, a flash memory, or an FD, or from an external storage medium via a network to the client computer, the present invention can be applied.

[0147] As mentioned above, it is needless to say that an object of the present invention is achieved even if a computer (or, a CPU or an MPU) of a system or an apparatus reads and executes a program code that is stored in a storage medium by supplying a storage medium recording a program code of software, which achieves a function of the above-described embodiment, to a system or an apparatus, or by downloading the program code from an external server (not shown).

[0148] In this case, the program code itself of the job account server or job account client program, or the like that is read from the storage medium achieves a new function of the present invention, and the storage medium that stores the program code constitutes the present invention. As a storage medium for supplying the program code, it is possible to use, for example, a floppy disk, a hard disk, an optical disk, a magneto-optical disk, a DVD, a CD-ROM, a magnetic tape, a nonvolatile memory card, ROM, EEPROM, and the like.

[0149] In addition, it is needless to say that it includes the case that not only a function of the above-described embodiment is achieved by a computer executing the read program code, but also a function of the above-described embodiment is achieved by an OS (operating system) or the like running on a computer and performing part or all of actual processing on the basis of an instruction of the program code. Moreover, it is needless to say that it includes the case that a CPU or the like that is provided in a feature expansion board or a feature expansion unit performs part or all of actual processing on the basis of an instruction of the program code after a program code, which is transmitted from an external predetermined server apparatus (not shown) via a network or is read from a storage medium, is written in memory provided in a feature expansion board inserted into the computer or a feature expansion unit connected to the computer, and a function of the above-described embodiment is achieved.

[0150] According to the present invention, it is possible to collect the acquired job information efficiently and without increasing a load in a wide range or plural bases by controlling whether all job information, managed by the management apparatus concerned, is acquired from the management apparatus managing information on the data processing, performed by a device, as job information, or,

acquiring additional job information from among the job information managed by the management apparatus concerned. In addition, since it is possible to acquire all job information managed by the management apparatus by having acquired the differential job information even if unknown information is included, it is possible to correct a mismatch when aggregation is performed on the basis of the job information.

[0151] Furthermore, the above-described differential acquisition means can collect job information collected in a wide range or plural bases efficiently and without increasing a load by acquiring additional job information after job information was acquired last time.

[0152] Moreover, it is possible to set the time of each management apparatus acquiring job information and to prevent processing loads and communication loads from concentrating temporarily because of the concentration of job information acquisition, by making a display unit display a screen for setting at least one of the time of acquiring all or additional job information, an interval of acquiring all job information, and the time of acquiring all job information.

[0153] According to the present invention, it is possible to flexibly manage job information without discarding information where a mismatch occurs by acquiring job information, managed by a management apparatus, from the concerned management apparatus managing information on the data processing, performed by a device, as job information, determining whether a value of an item that the acquired job information shows is new, and modifying a value of the above-described item to a value showing uncertainty when it is determined that it is new.

[0154] In addition, it is possible to flexibly correspond to information where a mismatch occurs without acquiring all the past information every time and to clarify a user, a device, or the like, which is newly added, as an unknown one, by determining that a value of an item that the job information that is acquired shows is new when the value of the item that the job information that is acquired shows when additional job information is acquired from among job information managed by a management apparatus is not included in values of the item that the job information that is acquired shows when all job information managed by the management apparatus is acquired.

[0155] Furthermore, it is possible to flexibly correspond to information where a mismatch occurs and to perform aggregation corresponding to user's intention, by making the user set whether job information where a value of an item shows uncertainty is used when aggregation based on the job information that is managed is performed.

[0156] It is possible to flexibly correspond to information where a mismatch occurs and to perform information management corresponding to user's intention, by making the user set whether information showing uncertainty is displayed when aggregation based on the job information that is managed is performed and aggregation result is displayed.

What is claimed is:

1. A management system that includes a management apparatus managing information on data processing performed by a device as job information, and an integrated management apparatus acquiring and managing job information managed by a plurality of management apparatuses, wherein the integrated management apparatus comprises:

full acquisition means for acquiring all job information, managed by a management apparatus, from the management apparatus designated from among a plurality of management apparatuses; and

differential acquisition means for acquiring additional job information among job information, managed by a management apparatus, from the management apparatus designated from among a plurality of management apparatuses.

2. An information processing apparatus that can communicate with a management apparatus managing information on data processing, performed by a device, as job information and that acquires and manages job information managed by the management apparatus, comprising:

full acquisition means for acquiring all job information, managed by the management apparatus, from the management apparatus; and

differential acquisition means for acquiring additional job information among job information, managed by the management apparatus, from the management apparatus.

3. A management method for communicating with a management apparatus managing information on data processing, performed by a device, as job information, and acquiring and managing job information managed by the management apparatus, comprising:

a full acquisition step of acquiring all job information, managed by the management apparatus, from the management apparatus; and

a differential acquisition step of acquiring additional job information among job information, managed by the management apparatus, from the management apparatus.

4. The management method according to claim 3, wherein the differential acquisition step acquires job information that is added after job information was acquired last time.

5. The management method according to claim 3, wherein the management method makes a display unit display a screen for setting at least one of time of acquiring job information by said full acquisition step or said differential acquisition step, an interval of acquiring job information by said full acquisition step, and time of acquiring all job information by said full acquisition step.

6. The management method according to claim 3, wherein the data processing is printing processing based on print data transmitted from an information processing apparatus; and

wherein the job information includes at least one among job identification information for identifying data processing, information showing an application program that creates print data, information showing a number of logical pages of print data, information showing a document name of print data, information showing start time of data processing, information showing finish time of data processing, information showing paper size, information showing a number of printed physical pages, information showing whether duplex printing was performed, information showing a name of a user having requested data processing, information showing a number of delivery pages, information showing a

number of delivery sheets, information showing a device having performed data processing.

7. The management method according to claim 3, wherein the device is at least one of a printer, a copier, a compound machine, and a facsimile.

8. A management program for communicating with a management apparatus managing information on data processing, performed by a device, as job information, and acquiring and managing job information managed by the management apparatus, wherein the management program makes a computer execute:

a full acquisition step of acquiring all job information, managed by the management apparatus, from the management apparatus; and,

a differential acquisition step of acquiring additional job information among job information, managed by a management apparatus, from the management apparatus.

9. A recording medium that stores a management program for communicating with a management apparatus managing information on data processing, performed by a device, as job information, and acquiring and managing job information managed by the management apparatus, and can be read by a computer, wherein the management program makes a computer execute:

a full acquisition step of acquiring all job information, managed by a management apparatus, from the management apparatus; and

a differential acquisition step of acquiring additional job information among job information, managed by the management apparatus, from the management apparatus.

10. A management system that includes management apparatuses managing information on data processing performed by each device as job information, and an integrated management apparatus acquiring and managing job information managed by a plurality of management apparatuses, wherein the integrated management apparatus comprises:

acquisition means for acquiring job information, managed by the management apparatus, from the management apparatus;

determination means for determining whether a value of an item that job information acquired by the acquisition means shows is new; and

modification means for modifying a value of an item to a value, showing uncertainty, when it is determined by the determination means that the value is new.

11. An information processing apparatus that can communicate with a management apparatus managing information on data processing, performed by a device, as job information and that acquires and manages job information managed by the management apparatus, comprising:

acquisition means for acquiring job information, managed by a management apparatus, from the management apparatus;

determination means for determining whether a value of an item that job information acquired by the acquisition means shows is new; and

modification means for modifying a value of an item to a value, showing uncertainty, when it is determined by the determination means that the value is new.

12. A management method for communicating with a management apparatus managing information on data processing, performed by a device, as job information, and acquiring and managing job information managed by the management apparatus, comprising:

an acquisition step of acquiring job information, managed by the management apparatus, from the management apparatus;

a determination step of determining whether a value of an item that the acquired job information shows is new; and

a modification step of modifying a value of the item to a value, showing uncertainty, when it is determined that the value is new.

13. The management method according to claim 12, further comprising a step of making a user set whether job information where a value of an item shows uncertainty is used when aggregation based on the job information that is managed is performed.

14. The management method according to claim 12, further comprising a step of displaying a screen for making a user set whether job information where a value of an item shows uncertainty is used when aggregation based on the job information that is managed is performed and aggregation result is displayed.

15. The management method according to claim 12, wherein the item is an item showing a device, or an item showing a user who requested data processing.

16. The management method according to claim 12, wherein the data processing is printing processing based on print data transmitted from an information processing apparatus; and

wherein the job information includes at least one among job identification information for identifying data processing, information showing an application program that creates print data, information showing a number of logical pages of print data, information showing a document name of print data, information showing start time of data processing, information showing finish time of data processing, information showing paper size, information showing a number of printed physical pages, information showing whether duplex printing was performed, information showing a name of a user having requested data processing, information showing a number of delivery pages, information showing a number of delivery sheets, information showing a device having performed data processing.

17. The management method according to claim 12, wherein the device is at least one of a printer, a copier, a compound machine, and a facsimile.

18. A management program for communicating with a management apparatus managing information on data processing, performed by a device, as job information, and acquiring and managing job information managed by the management apparatus, wherein the management program makes a computer execute:

an acquisition step of acquiring job information, managed by the management apparatus, from the management apparatus;

a determination step of determining whether a value of an item that job information acquired at the acquisition step shows is new; and

a modification step of modifying a value of an item to a value, showing uncertainty, when it is determined at the determination step that the value is new.

19. The management program according to claim 18, where the management method makes a computer execute:

a management step of managing job information acquired at the acquisition step;

an aggregation step of performing aggregation on the basis of job information managed at the management step; and

a setting step of making a user set whether job information, where a value of an item shows uncertainty, is used when aggregation is performed at the aggregation step.

20. The management program according to claim 19, wherein said setting step allows the user to set whether information indicating the uncertainty is displayed when results of the aggregation are displayed.

* * * * *